Hackathon 2022

Omer Dvora, Tomer Fried, Maya Glassman, Noga Kril

Data cleaning process and Challenges

- **Filling NULL values:** around **40%** of the values of the train data were missing. We made different decisions per feature according to the data and its meaning.
- **Categorizing unique entries of data:** we had to group together values such as: ++, Pos, חיובי, etc.
- **De-duplication of rows:** we checked for the reason for the duplication and found that multiple rows of the same patient varied only in the doctor that submitted the information, and the Clinique visited. We created dummy variables for these features and joined the information into a single row per patient.
- **Creating features based on medical understanding of the data:** for example, we learned about "TNM staging system"[1] (globally recognized standard for classifying the extent of spread of cancer) and created features that corresponds with this system (Our column "tumor_size" = size or direct extent of the primary tumor based on TNM tags from the data).
- **Categorical to numerical features:** we used here different methods, "one-hot encoding", value frequency etc.

Model selection

- **Multi-classification (Task 1):** the main challenge we faced was how to perform classification, since each sample can get up to 3 labels. We decided to train 11 independent binary classifiers and create combined prediction for each sample based on all the classifiers predictions. The independent binary classifiers are decision trees. This method is called "one vs rest".
- **XGBoost (Task 2):** we decided to use ensemble "XGBoost" to predict the tumor size for each sample. We saw that this model was used in many similar usecases (for example, kaggel).
- **K-Fold:** In addition, to make sure we're not overfitting and test the model's ability to predict new data, we have implemented K-Fold cross-validation on the model. As a

[1] https://en.wikipedia.org/wiki/TNM_staging_system

result, the loss was higher (Because the train set was smaller) but the average of the validations' loss was still close to our predictions loss without K-Fold.

- **Splitting the data and training:** the data contained multiple rows for each patient, and the rows were almost identical. Therefore, we split our train, development, and test set such that there would be no overlap of patients between the datasets. Also, we aggregated each patient's rows into single row (using mean over values) in order to avoid overfitting the train data.

Results

- Our model succeeded in identifying spread of metastases but not necessary the metastases' location. This was relatively easy, as a quick view of the data can tell whether the cancer is in progress. But, to diagnose the location requires better medical understanding of the data and the relationships between features.

	Total diagnosed **metastases** in test data	Total predicted **metastases** **Not necessary correct location**
numbers	**402**	**164**
%	**100%**	**41%**

- Our model succeeded better on the second task. We assume that it is because the data includes features directly related to the tumor size.

Suggestions

- **Advanced models:** For Task 1, we could use complex models such as ensembles for the independent classifiers. In addition, we could have tried different approach for the multi-classification. For example, it is possible to transform the problem to multi-classes problem, by creating one binary classifier for every label combination present in the training set[2].
- **Correlation:** after Pre-Processing, we have created a correlation matrix between all features and labels. We did not find any labels which were highly correlated with a feature but found features that are correlated with each other. These correlations might present collinearity and therefore reduce the quality of the predictions. We

[2] https://en.wikipedia.org/wiki/Multi-label_classification

would have liked to dig dipper on these relationships to better understand the data and maybe produce additional features.